Neighbor embedding XOM for dimension reduction and visualization

Bunte, Kerstin; Hammer, Barbara; Villmann, Thomas; Biehl, Michael; Wismueller, Axel

Published in:
Neurocomputing

DOI:
10.1016/j.neucom.2010.11.027

Publication date:
2011

Neurocomputing 74 (2011) 1340–1350

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Neighbor embedding XOM for dimension reduction and visualization

Kerstin Bunte [a,b,c,*,1], Barbara Hammer [d], Thomas Villmann [e], Michael Biehl [a], Axel Wismüller [b,c,f,1]

[a] Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, 9700AK Groningen, The Netherlands
[b] Department of Radiology, University of Rochester, 601 Elmwood Avenue, Rochester, NY 14642-648, USA
[c] Department of Biomedical Engineering, University of Rochester, 601 Elmwood Avenue, Rochester, NY 14642-648, USA
[d] Bielefeld University, CITEC, Universitätsstraße 23, 33615 Bielefeld, Germany
[e] Department of Mathematics, University of Applied Sciences Mittweida, Germany
[f] Department of Radiology, University of Munich, Klinikum Innenstadt, Ziemssenstr. 1, 80336 Munich, Germany

ARTICLE INFO

Available online 19 February 2011

Keywords:
Dimension reduction
Visualization
Divergence optimization
Nonlinear embedding
Exploratory Observation Machine
Stochastic neighbor embedding

ABSTRACT

We present an extension of the Exploratory Observation Machine (XOM) for structure-preserving dimensionality reduction. Based on minimizing the Kullback–Leibler divergence of neighborhood functions in data and image spaces, this Neighbor Embedding XOM (NE-XOM) creates a link between fast sequential online learning known from topology-preserving mappings and principled direct divergence optimization approaches. We quantitatively evaluate our method on real-world data using multiple embedding quality measures. In this comparison, NE-XOM performs as a competitive trade-off between high embedding quality and low computational expense, which motivates its further use in real-world settings throughout science and engineering.

1. Introduction

Various dimension reduction techniques have been introduced based on different properties of the original data to be preserved. The spectrum ranges from linear projections of original data, such as in principal component analysis (PCA) or classical multidimensional scaling (MDS) [1] to a wide range of locally linear and nonlinear approaches, such as Isomap [2,3], locally linear embedding (LLE) [4], local linear coordination (LLC) [5], or charting [6,7]. Stochastic neighbor embedding (SNE) [8] and t-distributed SNE (t-SNE) [9] approximates the probability distribution in the high-dimensional space, defined by neighboring points, with their probability distribution in a lower-dimensional space. Other methods aim at the preservation of the classification accuracy in lower dimensions and incorporate the available label information for the embedding, e.g. linear discriminant analysis (LDA) [10] and generalizations of it [11], extensions of the self-organizing map (SOM) [12] incorporating class labels [13] and limited rank matrix learning vector quantization (LiRaM LVQ) [14,15]. For a comprehensive review on nonlinear dimensionality reduction methods, we refer to [16]. For an up-to-date overview on current developments in the field, we refer to the recent overview publication [17].

Recently, a novel computational approach to topology learning has attracted attention for advanced data processing: The Exploration Machine (Exploratory Observation Machine, XOM) [18–21] (and references therein) systematically reverses the data-processing workflow in topology-preserving mappings. By consistently exchanging functional and structural components of topology-preserving mappings, XOM can be seen as a computational framework that computes graphical representations of high-dimensional observations by a strategy of self-organized model adaptation. Although simple and computationally efficient, XOM enjoys a surprising flexibility to simultaneously contribute to several different domains of advanced machine learning, scientific data analysis, and visualization. In particular, it supports both structure-preserving dimensionality reduction and data clustering.

As has been pointed out in the cited literature, there is no restriction whatsoever on the distance measures used in XOM. Specifically, among a large number of different distance measures even including non-metric distances, the possibility has been proposed to apply advanced divergence measures such as the Kullback–Leibler divergence and the Itakura–Saito distance within the XOM framework. This idea is in line with recent approaches to introduce alternative dissimilarity measures for data processing, such as Sobolev-distances or kernel based dissimilarity measures [16], approaches based on information theory using divergences for data processing, e.g. clustering [22–25], blind source separation [26], dimension reduction with MDS [1], or SNE.

* Corresponding author at: Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, 9700AK Groningen, The Netherlands. Tel.: +31 50 363 7049
E-mail address: k.bunte@rug.nl (K. Bunte).
URL: http://www.cs.rug.nl/~kbunte (K. Bunte).
[1] These authors contributed equally.

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.11.027

In this contribution, we extend and investigate the approach proposed in [27], called Neighbor Embedding XOM (NE-XOM) that builds upon the generalized Kullback–Leibler divergence as a dissimilarity measure between the neighborhood distributions of high-dimensional data and low-dimensional image vectors. The complexity of most nonlinear dimension reduction techniques grows quadratic with the number of points to embed. The aim of NE-XOM is to create a conceptual link between fast sequential online learning known from topology-preserving mappings and principled direct divergence optimization approaches, such as SNE and t-SNE. So it can be seen as a trade-off between low computational costs and high quality of the final embedding. The complexity is linear with the number of points and can be easily controlled by the user. Furthermore, prior knowledge and task specific requirements can be incorporated to the embedding result.

We will describe the basic algorithms XOM, SNE and t-SNE and the NE-XOM extension in Sections 2–4. We discuss the parameters in Section 5 and furthermore we spend some words on the complexity in comparison with other techniques in Section 6, discuss the embedding results on two benchmark data sets in Section 7, and conclude in Section 8.

2. The exploratory observation machine

We briefly review the Exploratory Observation Machine (XOM) algorithm. For details, we refer to the literature [18,21].

XOM maps a finite number of data points $\boldsymbol{x}^i \in \mathbb{R}^D$ in observation space $\mathcal{X}$ to low-dimensional data points $\boldsymbol{y}^i \in \mathbb{R}^d$ in the embedding space $\mathcal{E}$. The assignment is $\boldsymbol{x}^i \to \boldsymbol{y}^i$ and typically $d \ll D$, e.g. $d = 2$, 3 for visualization purposes. The embedding space $\mathcal{E}$ is priorly equipped with a structure hypothesis $p(\boldsymbol{s})$, usually given by a distribution $p(\boldsymbol{s})$ of sampling vectors $\boldsymbol{s} \in \mathbb{R}^d$, which corresponds to the final structure the data is embedded. Essentially, this is a generalization of the prototypes as included in the self-organizing map (SOM). Reasonable choices for the sampling vectors $\boldsymbol{s}$ are [18]: the location on a regular lattice structure in $\mathbb{R}^d$ just as in SOM, the location at discrete positions $\mathbb{R}^d$ to represent a finite number of class centers, the sampling according to a mixture of Gaussians centered in $\mathbb{R}^d$ to represent a finite number of clusters, or the uniform sampling in a region of $\mathbb{R}^d$ to indicate that the visualization of the data should occupy the full projection space. Unlike SOM, XOM does not project the sampling vectors $\boldsymbol{s}$ (generalization of prototypes) to the data space, rather, it projects the data to the embedding space. Nevertheless, the sampling vectors define receptive fields by a decomposition into points mapped closest to the sampling vectors. An approximate back projection of the sampling vector can be defined as the best match input vector

$$\Psi(\boldsymbol{s}) = \boldsymbol{x}^i \quad \text{where } d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^i) \text{ is minimum.} \tag{1}$$

The images $\boldsymbol{y}^i$ are initialized randomly and adapted iteratively during the training triggered by the structure of the embedding space. All $\boldsymbol{y}^i$ are adapted into the direction of the actual $\boldsymbol{s}$ according to the distances between the best match input $\Psi(\boldsymbol{s})$ and their counterparts $\boldsymbol{x}^i$ in the observation space $\mathcal{X}$. For a given sampling vector $\boldsymbol{s}$ the adaptation rule is given by

$$\boldsymbol{y}^k := \boldsymbol{y}^k - \eta h_\sigma(d_{\mathcal{X}}(\Psi(\boldsymbol{s}),\boldsymbol{x}^k)) \frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}, \tag{2}$$

where $\eta > 0$ denotes the learning rate, $d_{\mathcal{X}}$ refers to the distance in the observation space, e.g. the Euclidean distance and

$$h_\sigma(t) = \exp(-t/2\sigma^2) \quad \text{with } \sigma > 0 \tag{3}$$

defines the neighborhood cooperation. In this way the projections $\boldsymbol{y}$ are arranged around the priorly chosen structure elements $\boldsymbol{s}$ such that image vectors are close to the same sampling vector if their corresponding data points $\boldsymbol{x}$ are neighbored in the data space.

2.1. Cost function

As the SOM, XOM in its original form does not correspond to a cost function. However, as proposed in [27], a variation following Heskes [28] by setting the best match input data vector to the average

$$\Psi(\boldsymbol{s}) = \boldsymbol{x}^i \quad \text{where } \sum_j h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i,\boldsymbol{x}^j)) d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j) \tag{4}$$

is minimum, leads to the cost function:

$$E_{\mathrm{XOM}} \sim \int \sum_i \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i} \cdot \sum_{j=1}^{N} h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i,\boldsymbol{x}^j)) \cdot d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j) p(\boldsymbol{s})\, d\boldsymbol{s}, \tag{5}$$

where δ denotes the Kronecker delta. The derivative of E_{XOM} with respect to $\boldsymbol{y}^k$, which can be found in Appendix A, yields the XOM learning rule given in Eq. (2). Thus, XOM tries to minimize the distortion of sampling vectors $\boldsymbol{s}$ and projections $\boldsymbol{y}^j$ whereby this term is weighted according to a Gaussian function depending on the distance of the inverse images $\Psi(\boldsymbol{s})$ and $\boldsymbol{x}^j$ in the data space.

3. Review of SNE and t-SNE

A recent and very powerful proposal for data visualization is SNE [8]. It aims in finding projections such that the pairwise neighborhood distributions of points in the data and embedding spaces are approximately the same measured by the Kullback–Leibler (KL) divergence. SNE defines the following conditional probabilities $p_{j|i}$, which define the probability that a data point $\boldsymbol{x}^i$ would have another data point $\boldsymbol{x}^j$ as its neighbor. These affinities can, e.g. be defined by

$$p_{j|i} = \frac{\exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^k\|^2/2\sigma_i^2)} \tag{6}$$

and

$$q_{j|i} = \frac{\exp(-\|\boldsymbol{y}^i - \boldsymbol{y}^j\|^2)}{\sum_{k \neq i} \exp(-\|\boldsymbol{y}^i - \boldsymbol{y}^k\|^2)}, \tag{7}$$

with $p_{i|i} = q_{i|i} = 0$ for the data points and low-dimensional counterparts $\boldsymbol{y}$ in latent space, respectively. Each bandwidth σ_i is chosen by a binary search, such that the entropy of the distribution over neighbors is roughly log$k$, which is called *perplexity* or effective number of neighbors.

SNE tries to match the distribution P_i defined over all data points $\boldsymbol{x}^i$ with Q_i, the distribution of their low-dimensional counterparts. The dissimilarity is measured by the Kullback–Leiber divergence, thus the cost function is given by

$$C = \sum_i \mathrm{KL}(P_i \| Q_i) = \sum_{ij} p_{j|i} \log\left(\frac{p_{j|i}}{q_{j|i}}\right). \tag{8}$$

An alternative to minimizing the sum of Kullback–Leibler divergences between conditional probabilities $p_{j|i}$ and $q_{j|i}$ is called symmetric SNE. In that variant symmetrized pairwise similarities $p_{ij} = p_{ji} = (p_{j|i} + p_{i|j})/2n$ for n data points are used. This has the main advantage of a simpler form of its gradient, which is faster to compute.

To face the so-called *crowding problem*, which usually occurs in high dimensions and is also referred to as curse of dimensionality, the t-distributed SNE (t-SNE) was introduced [9]. In that variant the probability distribution in the low-dimensional map is modeled with much heavier tails than a Gaussian to convert distances

into probabilities. This allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map. In t-SNE a Student t-distribution with one degree of freedom is used, resulting in probabilities q_{ij}:

$$q_{ij} = \frac{(1+\|\boldsymbol{y}^i-\boldsymbol{y}^j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\boldsymbol{y}^k-\boldsymbol{y}^l\|^2)^{-1}}. \tag{9}$$

The gradient of the Kullback–Leibler divergence between P and the Student-t based joint probability distribution Q (using Eq. (9)) is given by

$$\frac{\delta C}{\delta \boldsymbol{y}^i} = 4\sum_j (p_{ij}-q_{ij})\frac{\boldsymbol{y}^i-\boldsymbol{y}^j}{1+\|\boldsymbol{y}^i-\boldsymbol{y}^j\|^2} \tag{10}$$

Further details and the derivation of the gradient can be found in [9].

4. Neighbor embedding XOM with generalized Kullback–Leibler divergence

XOM, unlike SNE and many other embedding algorithms, exhibits the interesting property that it allows to impose a prior structure on the projection space, which is a property that can also be found in SOM. Like many other visualization techniques, SNE has a computational and memory complexity that is quadratic in the number of data points. The complexity of XOM can be easily controlled by the structure definition and is linear with the number of data points and the number of sampling vectors. We propose to combine the ideas of XOM with the concept of direct divergence optimization as proposed by SNE.

By means of the cost function (5) we are able to define new learning rules for the XOM algorithm based on the generalized KL divergence for not normalized positive measures p and q with $0 \le p,q \le 1$:

$$E_{\mathrm{GKL}}(p\|q) = \int \left[p(\boldsymbol{x})\log\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right] d\boldsymbol{x} - \int [p(\boldsymbol{x})-q(\boldsymbol{x})]\, d\boldsymbol{x}. \tag{11}$$

We consider the use of normalized and symmetrized probability densities as unnecessary restriction and define our concept in a more general way. In contrast to [29,30], however, we do not use the generalized KL divergence as a distance measure *within* the original or the embedding space, but as a dissimilarity measure *between* the two spaces. The cooperativity functions $h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i,\boldsymbol{x}^j))$ and $g_\gamma(d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j))$ used as positive measures, in the following abbreviated by h_σ^{ij} and g_γ^j, can be defined analogously to Eqs. (3) and (6). They model the neighborhoods in the original space and the embedding space. Following the ideas of t-distributed SNE (t-SNE) [9] the neighborhood function of the embedding space $g_\gamma(d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j))$ could be chosen as a heavy-tailed distribution, e.g. the Student-t-distribution equation (9). This should avoid the *crowding problem* [9], which may occur due to the volume difference between high-dimensional and low-dimensional spaces. In the following formulas we will give the most general definitions for flexible use of distances $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$ and positive measures h and g in the high-dimensional and low-dimensional space, as well as explicit examples of them.

Based on these settings, we define a novel cost function using the divergence E_{GKL} equation (11):

$$E_{\mathrm{KLX}} \sim \int \sum_i \delta_{\Psi^{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i} \sum_j \left[h_\sigma^{ij}\ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right] p(\boldsymbol{s})\, d\boldsymbol{s}, \tag{12}$$

where the best match data point for $\boldsymbol{s}$ is defined as

$$\Psi^{\mathrm{GKL}}(\boldsymbol{s}) = \boldsymbol{x}^i \text{ such that } \sum_j \left[h_\sigma^{ij}\ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right] \text{ is minimum.} \tag{13}$$

The derivative of this cost function (see Appendix B) with respect to the images $\boldsymbol{y}^k$ yields the online learning update rule for a given sampling vector $\boldsymbol{s}$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \eta\Delta\boldsymbol{y}^k,$$

$$\Delta\boldsymbol{y}^k = \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}_k}\left(1-\frac{h_\sigma^{ik}}{g_\gamma^k}\right). \tag{14}$$

In case of a Gaussian neighborhood function in the embedding space we refer to the learning rule

$$\Delta\boldsymbol{y}^k = \frac{1}{2\gamma^2}(h_\sigma^{ik}-g_\gamma^k)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k} \tag{15}$$

$$= \frac{\alpha_g}{2}(h_\sigma^{ik}-g_\gamma^k)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k} \tag{16}$$

with

$$g_\gamma^j = \exp\left(\frac{-d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)}{2\gamma^2}\right) \quad \text{and} \quad \alpha_g = \frac{1}{\gamma^2} \tag{17}$$

as Neighbor Embedding XOM (NE-XOM). For a t-distribution in the embedding space the learning rule reads

$$\Delta\boldsymbol{y}^k = \frac{1+\gamma}{2(\gamma+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k))}\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}(h_\sigma^{ik}-g_\gamma^k) \tag{18}$$

$$= \frac{\alpha_t}{2}(h_\sigma^{ik}-g_\gamma^k)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k} \tag{19}$$

with

$$g_\gamma^j = (1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)/\gamma)^{(-(\gamma+1)/2)} \tag{20}$$

and

$$\alpha_t = \frac{1+\gamma}{\gamma(1+d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)/\gamma)}. \tag{21}$$

This learning rule is in the following referred to as t-distributed NE-XOM (t-NE-XOM).

Algorithm 1. Simple code for NE-XOM.

Data: pairwise dissimilarities $d_{\mathcal{X}}(\boldsymbol{x}^i,\boldsymbol{x}^j)$
Input: number of iterations T, learning rate η, variances σ, γ, hypothesis $p(\boldsymbol{s})$
Output: low-dimensional images $\boldsymbol{y}$
begin
  compute neighborhood cooperations h_σ;
  for each data item $\boldsymbol{x}^k$, initialize image $\boldsymbol{y}^k$;
  for $t \ge T$ **do**
    draw random vector $\boldsymbol{s}$ from $p(\boldsymbol{s})$;
    find winner $\Psi^{\mathrm{GKL}}(\boldsymbol{s}) = \boldsymbol{x}^i$;
    compute neighborhood function g_γ;
    set $\boldsymbol{y}_{(t)}^k = \boldsymbol{y}^k - \eta(t)\Delta\boldsymbol{y}^k \ \forall k$ with
    $\Delta\boldsymbol{y}^k = \frac{\alpha_g}{2}(h_\sigma^{ik}-g_\gamma^k)\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}$
end

While the original XOM approach is based on attraction forces only (see Eq. (2)), the prototype update in Eq. (14) includes repulsion as well. This is due to the possibility of a change of the sign dependent on the fraction between the cooperativity function h and g. The XOM update emphasizes attraction and predominantly optimizes "continuity", such that small distances in $\mathcal{X}$ lead to small distances in $\mathcal{E}$. In contrast to the XOM adaptation rule, the NE-XOM adaptation is able to push less similar samples out of a region of a sampling vector, if the pulling force of the actual winning sample is weaker than the repulsive

force of the sampling vector. This also prevents image vectors of collapsing onto one point which is stated to be a problem in LLE [5]. Furthermore the parameter γ in the t-distributed version equation (18) can be used to control the granularity of the final embedding. Further information about the parameters can be found in Section 5.

4.1. NE-XOM without structure hypothesis

It is also possible to use this algorithm without a defined structure. One could simply change the definition of the sampling vectors, as inspired by [20,31], in such a way that they are selected in close proximity to the image vector positions.

Therefore, instead of choosing a sampling vector randomly according to a given distribution, we visit the images $\boldsymbol{y}$ sequentially and choose a sampling vector $\boldsymbol{s}^j = \tilde{\boldsymbol{y}}^j$ drawn from a distribution centered around the actual images $\boldsymbol{y}^j$. Examples could be a Gaussian, a localized uniform, or a t-distribution. In our experiments we denote the use of this variant with the term (*ws*) added to the method name. And we used a normal distribution with variance $\varsigma : \mathcal{N}(\boldsymbol{y}^j, \varsigma)$. The algorithm thus changes to:

1. Compute pairwise distances $d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j)$.
2. Randomly initialize "image vectors" $\boldsymbol{y}^i \in \mathcal{E}, i = 1, \ldots, N$ corresponding to each input vector $\boldsymbol{x}^i$.
3. Sweep through the randomized set $\boldsymbol{y}$, where one complete run is referred to as one epoch. For every $\boldsymbol{y}^j$, find a sampling vector drawn from a low variance distribution centered around $\boldsymbol{y}^j$. Subsequently, perform the update of all image vectors $\boldsymbol{y}$ following Eq. (14).
4. Another image vector is chosen and step 3 is repeated until a maximal number of epochs is reached.

The final positions of the vectors $\boldsymbol{y}$ represent the output of the algorithm. However, in this variant the NE-XOM is no longer bounded to a predefined structure, but creates its own similarity map. Note that in this variant the parameters have to be tuned carefully, so that the repulsive forces do not dominate the embedding. Furthermore the algorithm without structure hypothesis may be computationally more expensive if the number of data samples grows over the number of vectors, which would be used in a predefined structure.

5. Parameter setting

In this section we will shortly discuss the parameters and their influence on the final embedding of the NE-XOM algorithm. First, the dissimilarity measures $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$ of the observation and embedding space have to be chosen. In our experiments we used the squared Euclidean distance for both of them. Further one has to decide which neighborhood function g should be used in the embedding space. We show in this section the different behavior of the algorithm for two example cases: Gaussian and t-distribution. As in XOM, the sampling vectors $\boldsymbol{s}$ may be chosen to match application-specific user needs. They could for example be drawn from a uniform distribution, a Gaussian, several Gaussian clusters or they could build a regular grid of any shape. In our experiments we used triangular grids generated by DISTMESH [32]. The list of parameters, which are candidates for adaptation during training, contains:

- σ the variance of the neighborhood cooperation h in the observation space $\mathcal{X}$,
- γ the variance of the neighborhood cooperation g in the embedding space $\mathcal{E}$,
- η the learning rate in the gradient decent optimization.

The parameter σ resembles the variance of the neighborhood function from the original SOM and XOM algorithms and is decreased during training. In our experiments, we used a different σ_i for every data sample $\boldsymbol{x}^i$ so that an ε ball of variance σ_i would contain a fixed number n_k of neighbors. This ensures that also data samples in less dense regions have an effect on the embedding. All σ_i follow an annealing scheme of the n_k during training:

$$n_k(t) = n_k(t_1) \cdot \exp\left(-\frac{\log\left(\frac{n_k(t_1)}{n_k(t_{\text{end}})}\right)}{n_e} \cdot t \right), \tag{22}$$

with $n_k(t_1)$ and $n_k(t_{\text{end}})$ being the number of neighbors at the beginning and at the end of training and n_e the total number of epochs (sweeps through the sampling vectors). It is also possible to find appropriate σ_i by using the "perplexity" proposed for the SNE approach [8].

From Eq. (3) follows that the winner always gets the maximal attraction force of one. Therefore, it is quite possible that for a sampling vector always the same data point $\boldsymbol{x}_i$ becomes the winner $\Psi(\boldsymbol{s})$. To increase the probability that different samples become the winners to one sampling vector we adjusted the value of $h^{ii}_{\sigma_i}$ from the winner to $0.9 \cdot \max_{i \neq j}(h^{ij}_{\sigma_i})$. In this way more samples become winner and therefore more data samples influence the final embedding.

Fig. 1 shows the influence of the parameter γ on the repulsive forces g and the learning rate α_* in dependence of the distance between image and sampling vectors in the embedding space. Fig. 1a shows the influence of the value γ for the repulsive forces addressed by g and the learning rate factor α_g in case of a Gaussian used as neighborhood function in the embedding space. The repulsion forces which may cause instabilities can be easily suppressed by big distances between the sampling vectors and a small $\gamma \in [1,2]$. For bigger γ, the update would become vanishingly small. In this case, the γ can be fixed during training, while the learning rate η is decreased following an annealing scheme. One may also start with high repulsive forces denoted by a bigger value of γ and decrease it during training following an annealing scheme:

$$\gamma(t) = \gamma(t_1) \cdot \exp\left(-\frac{\log\left(\frac{\gamma(t_1)}{\gamma(t_{\text{end}})}\right)}{n_e} \cdot t \right), \tag{23}$$

with $\gamma(t_1)$ and $\gamma(t_{\text{end}})$ being the value of γ in the beginning and the end of the training. Note that in this case the learning rate η should be adapted inversely proportional to the factor α_g, so that the resulting learning rate factor $\eta \cdot \alpha_g$ is decreased during training.

The application of a t-distribution in the embedding space shows an interesting behavior of the update strength α_t in dependence of the distance $d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{y}^j)$ (see Fig. 1b). Here, with γ, the localization of the update in the embedding space can be controlled. A high value of γ ensures the same update strength for all samples. For lower values only samples in the direct neighborhood of the actual sampling vector are updated, see Fig. 2. With the parameter γ for the t-distribution we can control the granularity or level of detail in the final similarity map. The learning rate η is very limited in this case and it is fixed to one. The value of γ is decreased during training with a similar annealing scheme as Eq. (23).

In summary, the parameter which depends on the actual data set at hand is σ for the neighborhood function in the observation space $\mathcal{X}$. The other parameters like the sampling distribution $\boldsymbol{s}$ are dependent on the needs and preferences of the user, but not on the data itself. As in original XOM, prior knowledge may be integrated in the choice of the structure. The parameter γ for the cooperativity function in the embedding space is adjusted according to the choice of the structure hypothesis and the level of detail the user desires.

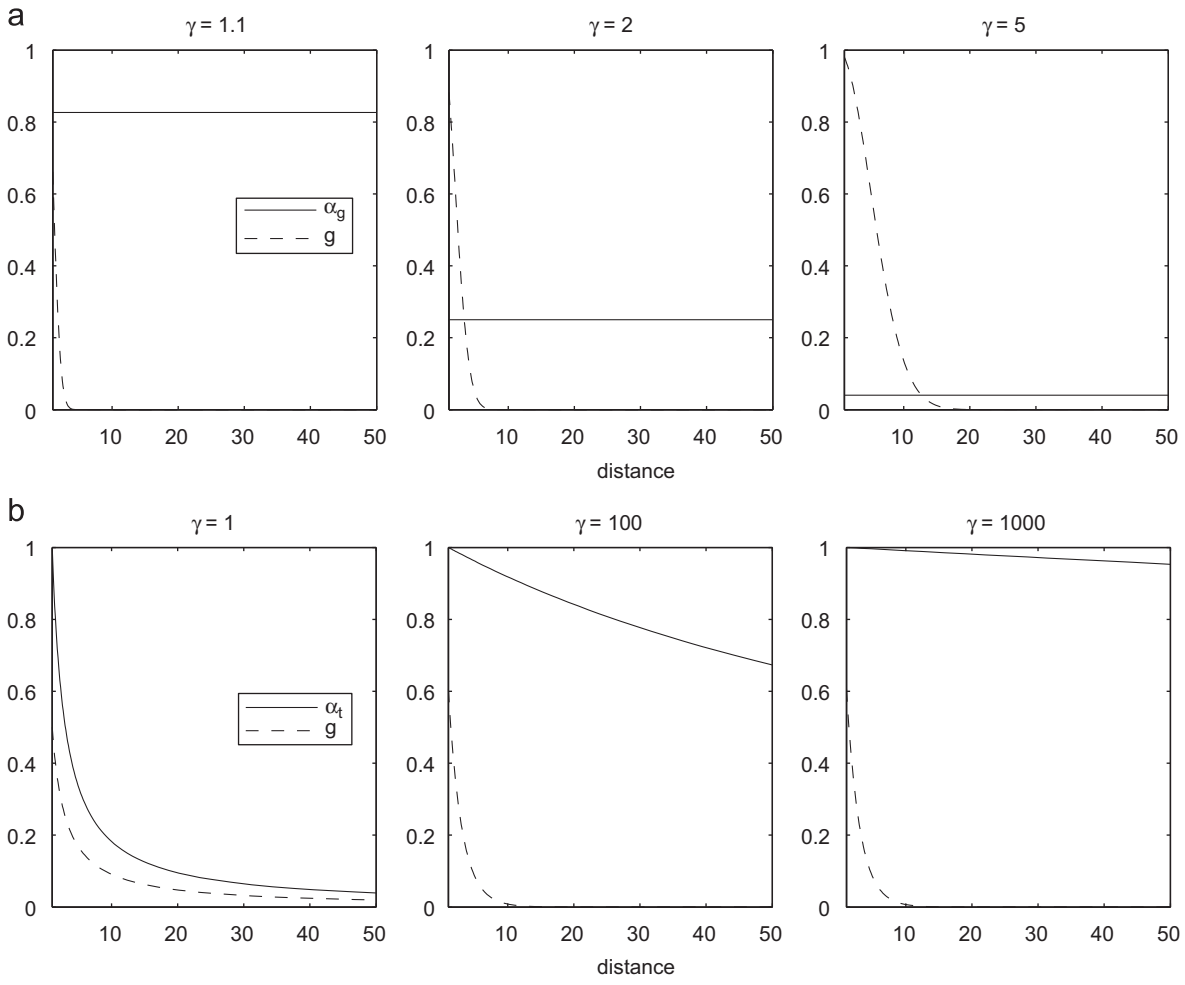

Fig. 1. Influence of the parameter γ on the repulsion forces g and the learning rate factor α_* in NE-XOM for given distances $d_\mathcal{E}$ in the embedding space. In (a) the neighborhood cooperation g is a Gaussian and α_g the resulting factor, see Eq. (17), which influences the learning rate η. In (b) g is given by Eq. (20), α_t is defined in Eq. (21). (a) Gaussian neighborhood cooperation function in the Embedding space and (b) t-distributed neighborhood cooperation function in the Embedding space.

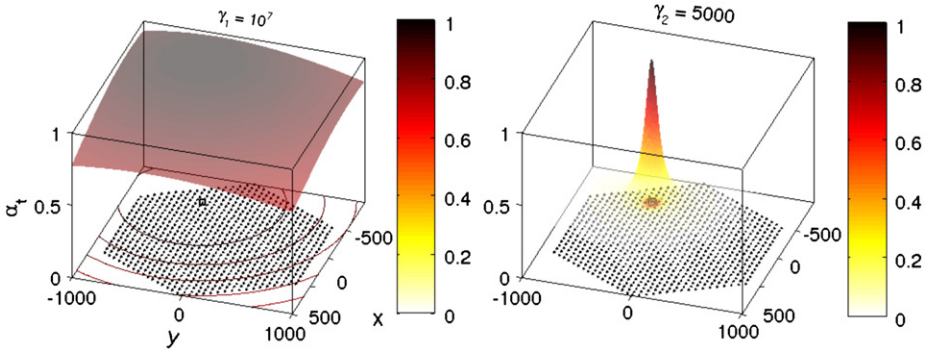

Fig. 2. The influence of the parameter γ for the learning rate factor α_t in t-NE-XOM using a t-distribution in the embedding space. The sampling vectors lie on a regular grid of hexagonal shape. For big values of γ, all image vectors are updated with nearly equal strength. With smaller values the update strength of image vectors outside the direct neighborhood of a sampling vector is suppressed.

6. Complexity

The complexity of the structure variant of NE-XOM depends on the dimension d of the embedding space $\mathcal{X}$, the number of samples to embed N, the number of sampling vectors S (which is usually much smaller than N) and the number of epochs (sweeps through the sampling set). So, every epoch calculations of the complexity $\mathcal{O}(dNS)$ have to be computed.

Fig. 3 shows the computational advantage of the simplest variant of NE-XOM in dependence of the number of data points to be embedded. For SNE and NE-XOM we used the same number of 1000 iterations and run the simulation on the same machine and all of them were matlab implementations. Most of the proposed dimension reduction techniques show at least quadratic complexity with the number of points to process. In those methods, the computation of the pairwise distances of the image vectors is

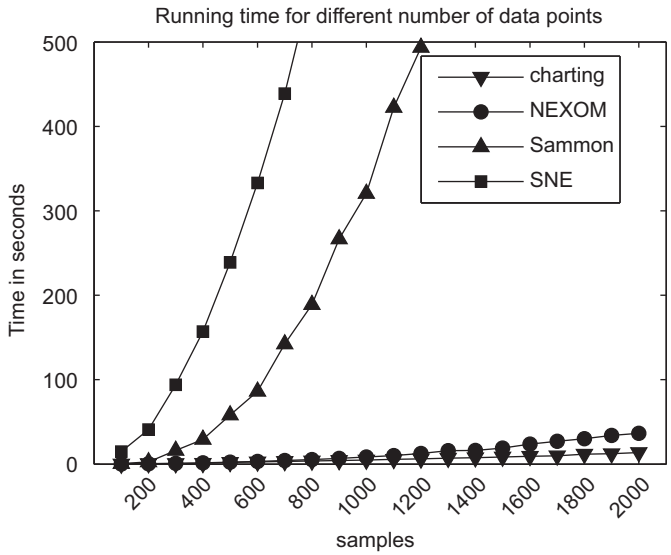

Fig. 3. The running time of different dimension reduction methods depending on the number of samples to embed.

necessary in every iteration. The structure variant of NE-XOM on the other hand only requires the computation of the distance of the image vectors to a given sampling vector in each iteration. Thus, for a sweep through the sampling set (one epoch) the complexity is dependent on the number of sampling vectors and the number of points, which is less than quadratic, if the number of sampling vectors is smaller than the data set size.

7. Experiments

In this section we show the results of different versions of two-dimensional NE-XOM on three exemplary real-world data sets. We compare some conventional quality measures, like Sammon's stress (Sammon) [33], Spearmans and Pearsons correlation (ρ_s and ρ_p) [34] as well as the nearest neighbor error ($\mathcal{E}_{NN}$ the error of the kNN classifier with $k=1$) and the K-intrusion and K-extrusion measure recently proposed by Lee and Verleysen [35,36], on the embeddings. Some methods we compare display linear complexity with the number of points, namely PCA and charting [16]. Additionally, we compare the results to those obtained from t-SNE, which is widely accepted as a high quality state-of-the-art technique, although it exhibits higher complexity and is computationally more expensive than the other techniques.

Many recently proposed quality measures for dimension reduction are based on the ranks of K-ary neighborhoods. The intrusion/extrusion diagram is one of them. It uses the co-ranking matrix containing the joint histogram of the ranks ρ_{ij} and r_{ij} based on the distances $d^{ij}_{\mathcal{X}}$ and $d^{ij}_{\mathcal{E}}$ in the observation and embedding space. Where ρ_{ij} defines the rank for the original vectors $\boldsymbol{x}^i$ in the high-dimensional space and r_{ij} the rank of the low-dimensional counterparts, respectively. The rank error is defined as the difference $\rho_{ij}-r_{ij}$. As intrusion we name the event of a vector j intruding a K-ary neighborhood $\mathcal{N}_i^K$ of another sample i, which is observed by a positive rank error. On the other hand the extrusion comes along with a negative rank error. $Q(K)$ is designed to measure the overall quality of an embedding by counting the fractions of mild K-intrusion, mild K-extrusions and the fraction of vectors that keep the same rank. On the other hand $B(K)$, which is defined by the difference of mild K-intrusions and mild K-extrusions, indicates the behavior of a dimension reduction method. $B(K)>0$ denotes intrusive and $B(K)<0$ extrusive embeddings. For further details we refer to [36].

7.1. USPS digits

The USPS[2] data set from the UCI repository [37] consists of images of hand-written digits of a resolution of 16×16 pixels. We use the digits $\in\{0,1,2,3,4\}$, resulting in 5500 samples. The parameter settings of all reduction techniques were optimized for performance, and on each parameter we performed 10 independent runs. For charting and t-SNE, we used the code provided by [5]. Charting yielded reasonable results for six analysers, while for t-SNE a perplexity of 45 provided good results. The other parameters were chosen according to default values provided by [5]. Some example embeddings are shown in Fig. 4 and the quality with different measures is shown in Fig. 5 and Table 1. The results of the NE-XOM algorithm were investigated using different variants: with and without structure hypothesis and with Gaussian and t-distribution in the embedding space respectively. The parameter settings can be found in Table 2.

The top left panel in Fig. 4 shows an example embedding of the NE-XOM algorithm with a Gaussian neighborhood function in the embedding space. The top right panel of Fig. 4 presents an example embedding of the t-NE-XOM algorithm using a t-distribution in the embedding space. Table 1 shows the results for Sammon's stress (Sammon), Spearmans and Pearsons correlation (ρ_s and ρ_p) for the different dimension reduction methods and the t-NE-XOM with structure using t-distribution. Two example results for embeddings without a structure hypothesis are shown in Fig. 6. The left side was achieved with NE-XOM(ws) using a Gaussian neighborhood and the right side is an example result of t-NE-XOM(ws).

From Fig. 5 and Table 1 it can be reasoned that the t-NE-XOM embedding can be identified as a competitive trade-off between high embedding quality and low computational expense. The different variants result in different behaviors of the embeddings: the incorporation of a Gaussian in the embedding space leads to similarity maps which preserve local neighborhoods, but prevents the image vectors of being projected onto each other. In addition, it forces image vectors to fill the whole structure. Using the t-distributed variant, the t-NE-XOM shows the ability of creating gaps between classes, and using a small γ the image vectors are not forced to spread in empty regions of the sampling space. In contrast to t-SNE (see Fig. 6) the (t-)NE-XOM embeddings with structure hypothesis (see Fig. 4) represent the different variances of the classes presented by the space they occupy in the embeddings. The digits equal to one are always confined to a small number of sampling vectors, whereas the 2's and 4's occupy a big region.

7.2. Relational data

As the NE-XOM algorithm depends on the topology of the observed data only, it can deal with pairwise distances as input. This is a property that NE-XOM directly inherits from the original XOM algorithm, which has been applied to the visualization of non-metric real-world data. These data sets are known as dissimilarity or relational data sets and they are often found in biological real-world problems, in which a data representation in vector form is not feasible.

As two examples we chose the Cat Cortex data set [38] preprocessed by Haasdonk [39] and Protein data [40]. The Cat Cortex originates from anatomic studies of cats' brains. This data set is given as a matrix containing the connection strength between 65 cortical areas splitted into four classes corresponding to four different regions of the cortex. The similarity matrix is symmetric but the triangle inequality does not hold. The Protein data contains the evolutionary distances of 226 globin proteins [40]. We use the

[2] United States Postal Service (U.S. Postal Service).

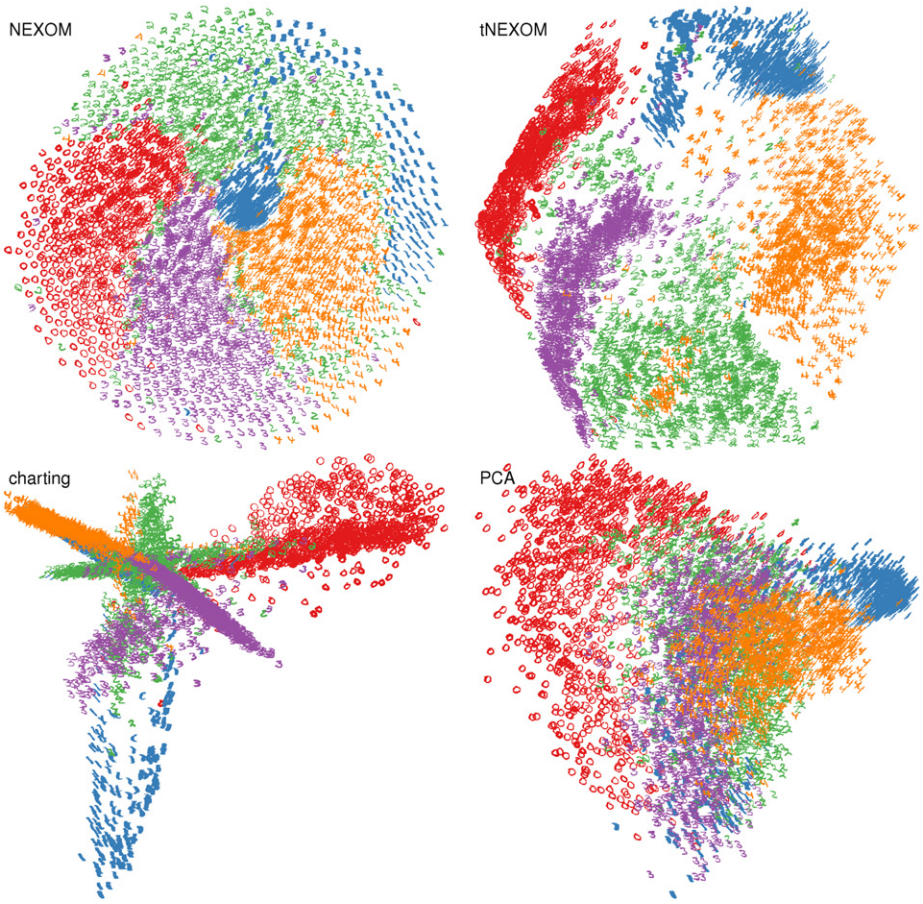

Fig. 4. Example embeddings of the USPS data set from different methods. From the upper left till lower right it shows: First, one result for the NE-XOM with a Gaussian in the embedding space and sampling vectors forming a regular circle ($\mathcal{E}_{NN} = 0.13$), second, one result of t-NE-XOM using a t-distribution and a regular sampling grid of hexagonal structure ($\mathcal{E}_{NN} = 0.05$), third, an example result of charting with six analyzers ($\mathcal{E}_{NN} = 0.26$), and last, the result of PCA ($\mathcal{E}_{NN} = 0.37$).

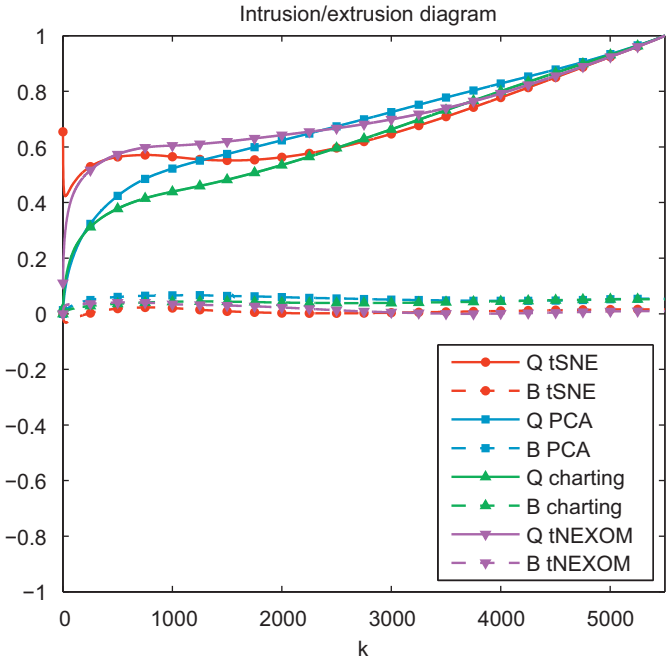

Fig. 5. Values of the overall quality Q and B as a function of the number of neighbors K.

Table 1
Different quality measures for USPS.

Method	t-NE-XOM	Charting	t-SNE
Sammon	**0.16** (0.0)	0.25 (0.1)	**0.16** (0.0)
ρ_s	**0.54** (0.0)	0.42 (0.1)	0.40 (0.1)
ρ_p	**0.57** (0.0)	0.43 (0.1)	0.44 (0.1)
$\mathcal{E}_{NN}$	0.06 (0.0)	0.29 (0.1)	**0.02** (0.0)

five classes proposed in [39]: HA, HB, MY, GG/GP and others. The class others combines small classes form the original data set and represents only a small fraction of the whole data set.

Fig. 7 shows two example embeddings of the relational data sets. We run the t-NE-XOM algorithm 10 times for each data set with random initialization of the image vectors. The embedding quality is measured by $Q(K)$ and the behavior with $B(K)$ and compared to those from t-SNE with varying perplexity. The mean values and standard deviation (std) of these measures is shown in Fig. 8. For t-SNE the best results were achieved with perplexity 25. The parameter setting for the t-NE-XOM for the Cat Cortex and for the Protein data can be found in Table 2.

The quality of the embeddings of t-NE-XOM and t-SNE is comparable. With the Cat Cortex data t-SNE shows bigger standard deviation regarding the random initialization and more extrusive behavior for small neighborhoods. For the protein data the quality

Table 2
Explicit parameter settings for the NE-XOM variants in the experiments.

Method	Structure hypothesis	Epochs	σ_i	γ
USPS				
NE-XOM	Triangular mesh, in the form of a circle, 562 $\boldsymbol{s}$	50	$\sigma_i(t_1) = \text{perpl. } 30$ $\sigma_i(t_{\text{end}}) = \text{perpl. } 3$	$\gamma = 1$
t-NE-XOM	Triangular mesh, in the form of a hexagon	500	Eq. (22), $n_k(t_1) = 3000$ $n_k(t_{\text{end}}) = 10$	Eq. (23), $\gamma(t_1) = 10^7$ $\gamma(t_{\text{end}}) = 5000$
NE-XOM(ws)	No hypothesis! $\boldsymbol{s}$ drawn from $\mathcal{N}(\boldsymbol{y}^j, 0.1)$	300	$\sigma_i(t_1) = \text{perpl. } 500$ $\sigma_i(t_{\text{end}}) = \text{perpl. } 5$	$\gamma = 1$
t-NE-XOM(ws)	No hypothesis! $\boldsymbol{s}$ drawn from $\mathcal{N}(\boldsymbol{y}^j, 10)$	300	$\sigma_i(t_1) = \text{perpl. } 500$ $\sigma_i(t_{\text{end}}) = \text{perpl. } 5$	Eq. (23), $\gamma(t_1) = 10^7$ $\gamma(t_{\text{end}}) = 0.1$
Catcortex				
t-NE-XOM	Triangular mesh, in the form of a hexagon, 48 $\boldsymbol{s}$	500	Eq. (22), $n_k(t_1) = 50$ $n_k(t_{\text{end}}) = 5$	Eq. (23), $\gamma(t_1) = 10^7$ $\gamma(t_{\text{end}}) = 1000$
Protein				
t-NE-XOM	Triangular mesh, in the form of a hexagon, 200 $\boldsymbol{s}$	500	Eq. (22), $n_k(t_1) = 200$ $n_k(t_{\text{end}}) = 5$	Eq. (23), $\gamma(t_1) = 10^7$ $\gamma(t_{\text{end}}) = 2000$

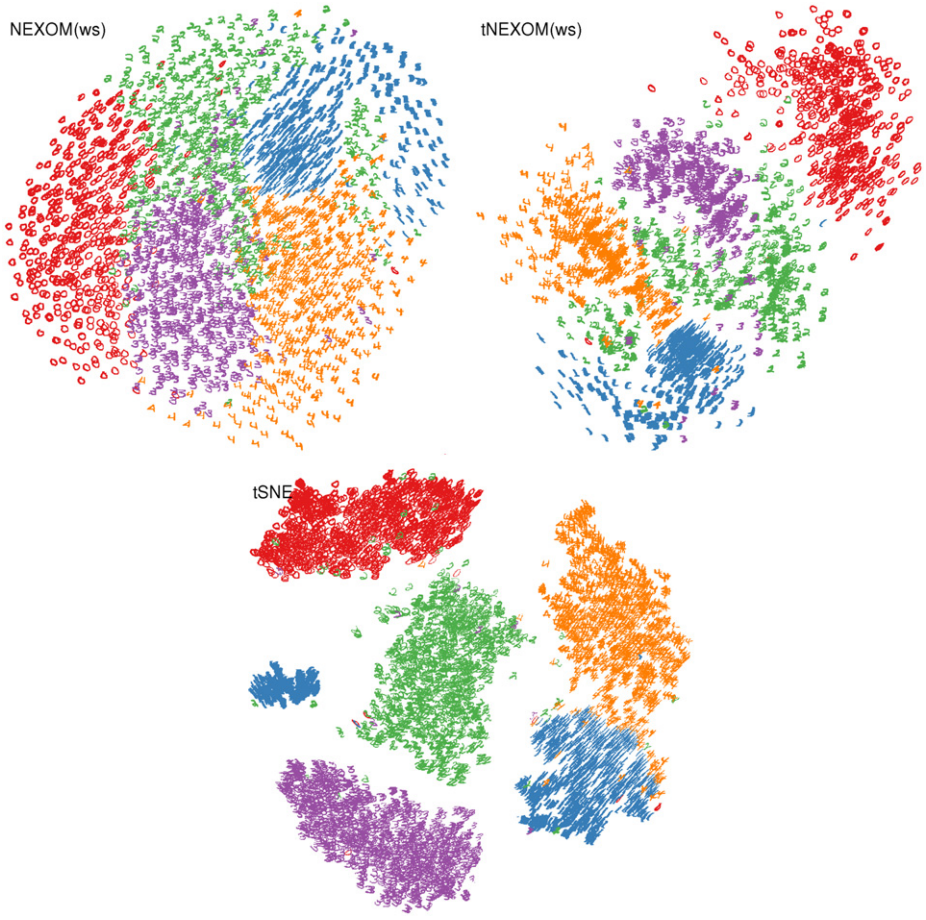

Fig. 6. Two example embeddings of the NE-XOM algorithm without a structure hypothesis and a t-SNE example embedding. For the left- and right-hand side, a Gaussian and a t-distribution were used in the embedding space, respectively.

measured by Q is higher with t-SNE and the embedding shows highly intrusive behavior. The t-NE-XOM embedding shows in this case extrusive behavior. This shows that despite the close relationship of SNE and NE-XOM even the behavior of the embeddings may vary a lot. The mean nearest neighbor error of the 10 t-NE-XOM embeddings is $\bar{\mathcal{E}}_{NN} = 0.13$ with standard deviation of 3% for the Cat Cortex and $\bar{\mathcal{E}}_{NN} = 0.08$ with std = 3% for the Protein data set.

8. Conclusion

In this contribution, we have introduced an extension of the Exploratory Observation Machine (XOM) for structure-preserving dimensionality reduction. Based on minimizing the Kullback–Leibler divergence of neighborhood functions in data and image space, NE-XOM creates a conceptual link between fast sequential online

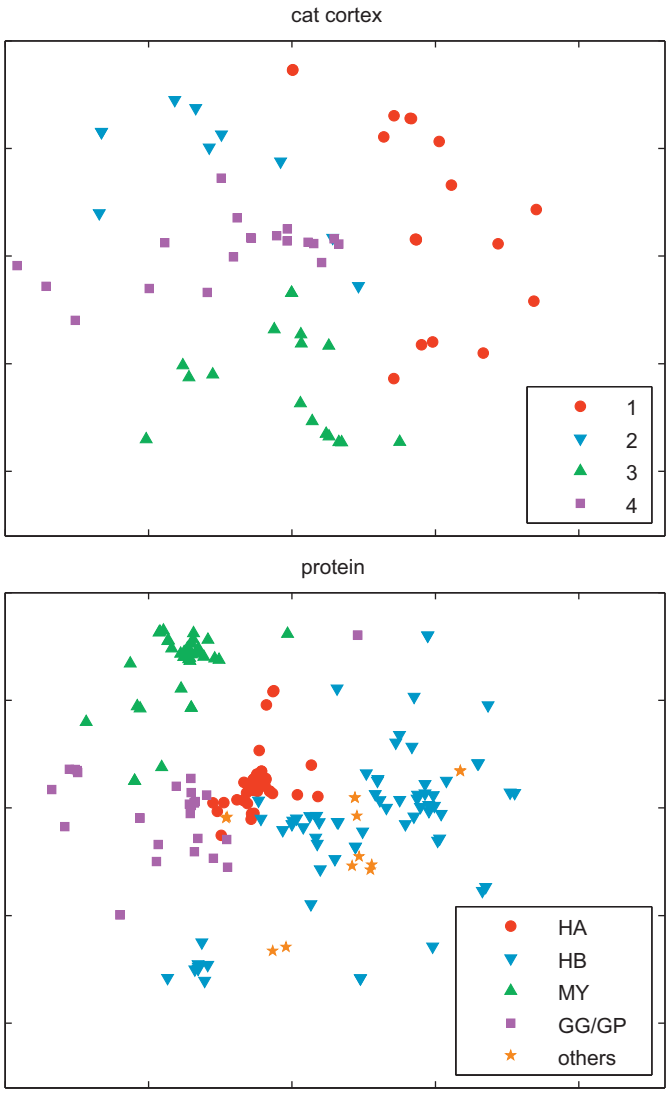

Fig. 7. Example embeddings of the Cat Cortex ($\mathcal{E}_{NN}=0.09$) and Protein ($\mathcal{E}_{NN}=0.04$).

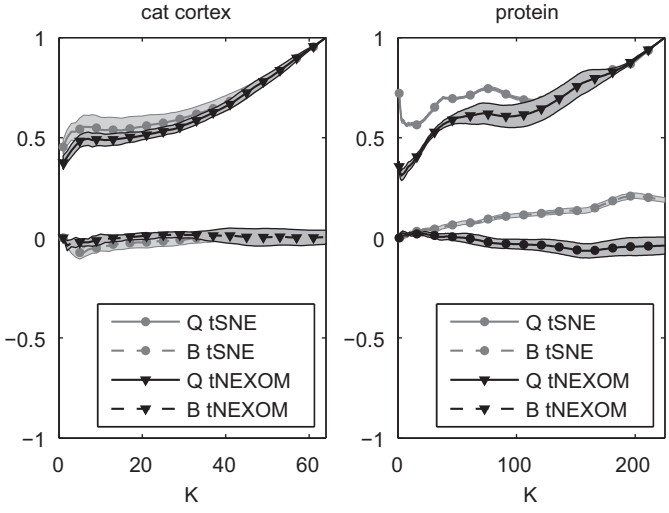

Fig. 8. The overall embedding quality $Q(K)$ and $B(K)$ for two relational data sets.

learning known from topology-preserving mappings and principled direct divergence optimization approaches, such as SNE and t-SNE. Quantitative comparative evaluation on benchmark data using multiple embedding quality measures identifies NE-XOM as a competitive trade-off between high embedding quality and low computational expense, which motivates its extended use in real-world settings throughout science and engineering. We have extended the algorithm to utilize different distributions, namely the Gaussian and the t-distribution following the ideas proposed in t-distributed SNE [9]. We have analyzed different variants of the NE-XOM algorithm with and without structure hypothesis and using different distributions, which offers high flexibility based on application needs. Finally, it allows the user to incorporate prior knowledge and to adapt the level of detail resolution. It enables the cooperation of prior knowledge and tuning of the level of detail the user desires. Future work will be addressing the extension of the algorithm to utilize different divergence measures.

Acknowledgments

This work was supported by the "Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)" under project code 612.066.620, from the United States National Center for Research Resources (NCRR), a component of the National Institutes of Health (NIH) and the NIH Roadmap for Medical Research under award number 5-28527, and by the Center of Emerging and Innovative Sciences (CEIS), a NYSTAR-designated Center for Advanced Technology. We are grateful for the stimulating discussions, example code and comments by John A. Lee.

Appendix A. Derivative of the XOM cost function

With the abbreviation $h_\sigma^{ij}=h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i,\boldsymbol{x}^j))$ we write the derivative of the cost function equation (5) with respect to an image vector $\boldsymbol{y}^k$:

$$\frac{\partial E_{XOM}}{\partial \boldsymbol{y}^k}=\int\sum_i\frac{\partial\delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k}\sum_j h_\sigma^{ij}d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)p(\boldsymbol{s})\,d\boldsymbol{s}+\int h_\sigma(d(\Psi(\boldsymbol{s}),\boldsymbol{x}^k))\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}p(\boldsymbol{s})\,d\boldsymbol{s}. \tag{A.1}$$

The second term yields the learning rule equation (2) while the first term vanishes due to the following considerations: we use the shorthand notation

$$\Phi(\boldsymbol{x}^i,\boldsymbol{s})=\sum_j h_\sigma^{ij}d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j). \tag{A.2}$$

Then, the Kronecker delta can be expressed as

$$\delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}=H\left(\sum_k H(\Phi(\boldsymbol{x}^i,\boldsymbol{s})-\Phi(\boldsymbol{x}^k,\boldsymbol{s}))-n+0.5\right), \tag{A.3}$$

where H denotes the Heaviside function and n denotes the number of data points $\boldsymbol{x}^i$. The derivative of H is given by the delta function δ which is symmetric and non-vanishing only for input zero. Hence the first term of Eq. (A.1) vanishes:

$$\begin{aligned}&\int\left[\sum_i\frac{\partial\delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k}\Phi(\boldsymbol{x}^i,\boldsymbol{s})\right]p(\boldsymbol{s})\,d\boldsymbol{s}\\&=\int\sum_i\delta\left(\sum_l H(\Phi(\boldsymbol{x}^i,\boldsymbol{s})-\Phi(\boldsymbol{x}^l,\boldsymbol{s}))-n+0.5\right)\\&\quad\cdot\sum_l\delta(\Phi(\boldsymbol{x}^i,\boldsymbol{s})-\Phi(\boldsymbol{x}^l,\boldsymbol{s}))\cdot(h_\sigma^{ik}-h_\sigma^{lk})\\&\quad\cdot\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{x}^k)}{\partial \boldsymbol{y}^k}\sum_j h_\sigma^{ij}d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)p(\boldsymbol{s})\,d\boldsymbol{s},\end{aligned} \tag{A.4}$$

$$\int \left[\sum_i \frac{\partial \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k} \Phi(\boldsymbol{x}^i,\boldsymbol{s})\right] p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$= \int \left[\sum_{ilj} \delta\left(\sum_{l'} H(\Phi(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)\right.$$
$$\cdot \delta(\Phi(\boldsymbol{x}^j,\boldsymbol{s}) - \Phi(\boldsymbol{x}^l,\boldsymbol{s})) \cdot h_\sigma^{ik} h_\sigma^{ij} \cdot d_\mathcal{E}(\boldsymbol{s},\boldsymbol{y}^j)$$
$$- \sum_{ilj} \delta\left(\sum_{l'} H(\Phi(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\left.\cdot \delta(\Phi(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi(\boldsymbol{x}^i,\boldsymbol{s})) \cdot h_\sigma^{ik} h_\sigma^{lj} d_\mathcal{E}(\boldsymbol{s},\boldsymbol{y}^j)\right] \cdot \frac{\partial d_\mathcal{E}(\boldsymbol{s},\boldsymbol{x}^k)}{\partial \boldsymbol{y}^k} p(\boldsymbol{s})\, d\boldsymbol{s}, \tag{A.5}$$

$$\int \left[\sum_i \frac{\partial \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k} \Phi(\boldsymbol{x}^i,\boldsymbol{s})\right] p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$= \int \left[\sum_{il} \delta\left(\sum_{l'} H(\Phi(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)\right.$$
$$\cdot \delta(\Phi(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi(\boldsymbol{x}^l,\boldsymbol{s})) \cdot h_\sigma^{ik} \Phi(\boldsymbol{x}^i,\boldsymbol{s})$$
$$- \sum_{il} \delta\left(\sum_{l'} H(\Phi(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\left.\cdot \delta(\Phi(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi(\boldsymbol{x}^l,\boldsymbol{s})) \cdot h_\sigma^{ik} \Phi(\boldsymbol{x}^l,\boldsymbol{s})\right] \cdot \frac{\partial d_\mathcal{E}(\boldsymbol{s},\boldsymbol{x}^k)}{\partial \boldsymbol{y}^k} p(\boldsymbol{s})\, d\boldsymbol{s} = 0. \tag{A.6}$$

Appendix B. Derivative of the NE-XOM cost function

With the abbreviation $h_\sigma^{ij} = h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i,\boldsymbol{x}^j))$ and $g_\gamma^j = g_\gamma(d_\mathcal{E}(\boldsymbol{s},\boldsymbol{y}^j))$ we write the derivative of the cost function equation (12) with respect to an image vector $\boldsymbol{y}^k$:

$$\frac{\partial E_{\mathrm{KLX}}}{\partial \boldsymbol{y}^k} \sim \int \sum_i \frac{\partial \delta_{\Psi^{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k} \sum_j \left(h_\sigma^{ij} \ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$+ \int \sum_i \delta_{\Psi^{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i} \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k} \left(1 - \frac{h_\sigma^{ik}}{g_\gamma^k}\right) p(\boldsymbol{s})\, d\boldsymbol{s}, \tag{B.1}$$

with $\Psi^{\mathrm{GKL}}(\boldsymbol{s})$ defined in Eq. (13). The latter term yields the learning rule. The first term vanishes, as can be seen as follows: We use the shorthand notation

$$\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) = \sum_j \left(h_\sigma^{ij} \ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right). \tag{B.2}$$

Then the best match input point can be expressed as

$$\delta_{\Psi^{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i} = H\left(\sum_k H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^k,\boldsymbol{s})) - n + \frac{1}{2}\right). \tag{B.3}$$

Hence the additional first term of Eq. (B.1) vanishes, because of the following:

$$\int \sum_i \frac{\partial \delta_{\Psi^{\mathrm{GKL}}(\boldsymbol{s}),\boldsymbol{x}^i}}{\partial \boldsymbol{y}^k} \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$= \int \sum_i \delta\left(\sum_l H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\cdot \sum_l \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) \cdot [(g_\gamma^k - h_\sigma^{ik}) - (g_\gamma^k - h_\sigma^{lk})] \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k} \tag{B.4}$$

$$\sum_j \left(h_\sigma^{ij} \ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$= \int \sum_{ilj} \delta\left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) \cdot (g_\gamma^k - h_\sigma^{ik}) \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k}$$
$$\cdot \left(h_\sigma^{ij} \ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$- \int \sum_{ilj} \delta\left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})) \cdot (g_\gamma^k - h_\sigma^{ik}) \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k}$$
$$\cdot \left(h_\sigma^{lj} \ln\left(\frac{h_\sigma^{lj}}{g_\gamma^j}\right) - h_\sigma^{lj} + g_\gamma^j\right) p(\boldsymbol{s})\, d\boldsymbol{s} \tag{B.5}$$

$$\sum_j \left(h_\sigma^{ij} \ln\left(\frac{h_\sigma^{ij}}{g_\gamma^j}\right) - h_\sigma^{ij} + g_\gamma^j\right) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$= \int \sum_{il} \delta\left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s})) \cdot (g_\gamma^k - h_\sigma^{ik}) \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k} \cdot \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s}) p(\boldsymbol{s})\, d\boldsymbol{s}$$
$$- \int \sum_{il} \delta\left(\sum_{l'} H(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^{l'},\boldsymbol{s})) - n + \frac{1}{2}\right)$$
$$\cdot \delta(\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) - \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})) \cdot (g_\gamma^k - h_\sigma^{ik}) \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}^k} \cdot \Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) p(\boldsymbol{s})\, d\boldsymbol{s} = 0, \tag{B.6}$$

because of the symmetry of δ and the fact that δ is nonvanishing only if $\Phi^{\mathrm{N}}(\boldsymbol{x}^l,\boldsymbol{s}) = \Phi^{\mathrm{N}}(\boldsymbol{x}^i,\boldsymbol{s})$.

References

[1] P.L. Lai, C. Fyfe, Bregman divergences and multi-dimensional scaling, in: 15th International Conference on Neuro-Information Processing: (ICONIP), Revised Selected Papers, Part II, Springer-Verlag, Auckland, New Zealand, 2008, pp. 935–942 doi: 10.1007/978-3-642-03040-6_114.

[2] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323 doi: 10.1126/science.290.5500.2319.

[3] V. De Silva, J. B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, in: Advances in Neural Information Processing Systems vol. 15, 2003, pp. 705–712. doi: 10.1.1.9.3407.

[4] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326 doi: 10.1126/science.290.5500.2323.

[5] L.J.P. van der Maaten, E.O. Postma, H.J. van den Herik, Dimensionality reduction: a comparative review, Technical Report TiCC-TR 2009-005, Tilburg University, October 2009.

[6] M. Brand, Charting a manifold, Technical Report 15, Mitsubishi Electric Research Laboratories (MERL), 2003.

[7] Y. Teh, S. Roweis, Automatic alignment of local representations, in: Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 841–848.

[8] G. Hinton, S. Roweis, Stochastic neighbor embedding, in: Neural Information Processing Systems 15 (NIPS), MIT Press, 2003, pp. 833–840.

[9] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.

[10] K. Fukunaga, Introduction to Statistical Pattern Recognition, Computer Science and Scientific Computing Series, second ed., Academic Press, 1990.

[11] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12 (10) (2000) 2385–2404.

[12] T. Kohonen, Self-Organizing Maps, third ed., Springer, Berlin, Heidelberg, New York, 2001.

[13] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, W. Hermann, Fuzzy classification by fuzzy labeled neural gas, Neural Networks 19 (6–7) (2006) 772–779.

[14] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, M. Biehl, Discriminative visualization by limited rank matrix learning, Technical Report MLR-03-2008, Leipzig University, 2008.

[15] K. Bunte, B. Hammer, A. Wismüller, M. Biehl, Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data, Neurocomputing 73 (7–9) (2010) 1074–1092 doi: 10.1016/j.neucom.2009.11.017.

[16] J. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, first ed., Springer, 2007.

[17] A. Wismüller, J.-A. Lee, M. Verleysen, M. Aupetit, Recent advances in nonlinear dimensionality reduction, manifold and topological learning, in: M. Verleysen (Ed.), 18th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2010, pp. 247–252.

[18] A. Wismüller, The exploration machine: a novel method for analyzing highdimensional data in computer-aided diagnosis, in: N. Karssemeijer, M. Giger (Eds.), Medical Imaging 2009: Computer-Aided Diagnosis, vol. 7260, SPIE, , 2009 72600G-72600G-7.

[19] A. Wismüller, A computational framework for nonlinear dimensionality reduction and clustering, in: J. Principe, R. Miikkulainen (Eds.), Advances in Self-Organizing Maps, Lecture Notes in Computer Science, vol. 5629, Springer, 2009, pp. 334–343.
[20] A. Wismüller, Exploration-organized morphogenesis (XOM) a general framework for learning by self-organization, in: Human and Machine Perception. Reports of the Institute for Phonetics and Speech Communication (FIPKM), vol. 37, 2001, pp. 205–239.
[21] A. Wismüller, Computational intelligence in biomedical imaging: multidimensional analysis of spatio-temporal patterns, Computer Science – R&D 26 (1–2) (2011) 15–37.
[22] A. Banerjee, S. Merugu, I.S. Dhillon, J. Ghosh, Clustering with Bregman divergences, Journal of Machine Learning Research 6 (2005) 1705–1749.
[23] T. Lehn-Schiøler, A. Hegde, D. Erdogmus, J.C. Principe, Vector-quantization using information theoretic concepts, Natural Computing 4 (2005) 39–51 doi: 10.1007/s11047-004-9619-8..
[24] E. Jang, C. Fyfe, H. Ko, Bregman divergences and the self organising map, in: IDEAL '08: 9th International Conference on Intelligent Data Engineering and Automated Learning, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 452–458 doi: 10.1007/978-3-540-88906-9_57.
[25] A. Hegde, D. Erdogmus, T. Lehn-Schiøler, Y. Rao, J. Principe, Vector-quantization by density matching in the minimum Kullback–Leibler divergence sense, in: IEEE International Conference on Neural Networks, vol. 1, 2004, pp. 105–109.
[26] M. Mihoko, S. Eguchi, Robust blind source separation by beta divergence, Neural Computation 14 (8) (2002) 1859–1886 doi: 10.1162/089976602760 128045.
[27] K. Bunte, B. Hammer, T. Villmann, M. Biehl, A. Wismüller, Exploratory observation machine (XOM) with Kullback–Leibler divergence for dimensionality reduction and visualization, in: M. Verleysen (Ed.), 18th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2010, pp. 87–92.
[28] T. Heskes, Energy functions for self-organizing maps, 1999.
[29] T. Villmann, S. Haase, Mathematical foundations of th generalization of t-SNE amd SNE for arbitrary divergences, Technical Report MLR-02-2010, Leipzig University, 2010.
[30] E. Mwebaze, P. Schneider, F. M. Schleif, S. Haase, T. Villmann, M. Biehl, Divergence based Learning Vector Quantization, in: M. Verleysen (Ed.), 18th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2010, pp. 247–252.
[31] J. A. Lee, C. Archambeau, M. Verleysen, Locally linear embedding versus Isotop, in: 11th European Symposium on Artificial Neural Networks (ESANN), 2003, pp. 527–534.
[32] P.-O. Persson, G. Strang, A simple mesh generator in matlab, SIAM Review 46 (2) (2004) 329–345.
[33] J. Sammon, A nonlinear mapping for data structure analysis, IEEE Transactions on Computers C 18 (1969) 401–409.
[34] J. Venna, Dimensionality reduction for visual exploration of similarity structures, Ph.D. Thesis, Helsinki University of Technology, 2007.
[35] J. A. Lee, M. Verleysen, Rank-based quality assessment of nonlinear dimensionality reduction, in: 16th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2008, pp. 49–54.
[36] J.A. Lee, M. Verleysen, Quality assessment of dimensionality reduction: rank-based criteria, Neurocomputing 72 (7–9) (2009) 1431–1443 doi: 10.1016/j.neucom.2008.12.017.
[37] A. Asuncion, D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998.
[38] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, K. Obermayer, Classification on pairwise proximity data, in: 1998 Conference on Advances in Neural Information Processing Systems II, MIT Press, Cambridge, MA, USA, 1999, pp. 438–444.
[39] B. Haasdonk, C. Bahlmann, Learning with distance substitution kernels, Pattern Recognition 3175 (2004) 220–227 doi: 10.1007/978-3-540-28649-3_27.
[40] H.T. Mevissen, M. Vingron, Quantifying the local reliability of a sequence alignment, Protein Engineering 9 (1996) 127–132.

Kerstin Bunte graduated at the Faculty of Technology at the University of Bielefeld, Germany, and joined the Institute of Mathematics and Computing Science of the University of Groningen, The Netherlands, in September 2007. Her recent work has focused on Machine Learning techniques, especially Learning Vector Quantisation and their usability in the field of image processing, supervised dimension reduction and visualization. Further information can be obtained from http://www.cs.rug.nl/~kbunte/

Barbara Hammer received her Ph.D. in Computer Science in 1995 and her venia legendi in Computer Science in 2003, both from the University of Osnabrueck, Germany. From 2000 to 2004, she was a leader of the junior research group "Learning with Neural Methods on Structured Data" at the University of Osnabrueck before accepting an offer as professor for Theoretical Computer Science at Clausthal University of Technology, Germany, in 2004. Several research stays have taken her to Italy, UK, India, France, and the USA. Her areas of expertise include various techniques such as hybrid systems, self-organizing maps, clustering, and recurrent networks as well as applications in bioinformatics, industrial process monitoring, or cognitive science. Most of her publications can be retrieved from http://www.in.tu-clausthal.de/~hammer

Thomas Villmann holds a diploma degree in mathematics and received his Ph.D. in Computer Science in 1996 and his venia legendi in the same subject in 2005, both from the University of Leipzig. From 1997 to 2009 he led the research group of computational intelligence of the clinic for psychotherapy at Leipzig University. Since 2009 he is a full professor for Technomathematics and Computational Intelligence at the University of Applied Science Mittweida, Germany. He is founding member of the German chapter of ENNS, member of the IEEE TaskforceCIS task force on Data Visualization and Data Analysis, and speaker of the working group "neural Networks" of the German Computer Science Society (GI). His research areas include a broad range of machine learning approaches like neural maps, clustering, classification, pattern recognition and evolutionary algorithms as well as applications in medicine, bioinformatics, satellite remote sensing and others.

Michael Biehl received a Ph.D. in Theoretical Physics from the University of Giessen, Germany, in 1992 and the venia legendi in Theoretical Physics from the University of Würzburg, Germany, in 1996. He is currently Associate Professor with Tenure in Computing Science at the University of Groningen, The Netherlands. His main research interest is in the theory, modelling and application of Machine Learning techniques. He is furthermore active in the modelling and simulation of complex physical systems. He has co-authored more than 100 publications in international journals and conferences; preprint versions and further information can be obtained from http://www.cs.rug.nl/~biehl/

Axel Wismüller studied medicine at the Technical University of Munich and the University of Regensburg, Germany, with study exchange programs in Switzerland and the USA (Yale University). He received his M.D. (Dr. med.) degree from the Technical University of Munich for a scientific thesis in neurology in 1992. He successfully passed the U.S. medical examinations ECFMG and FLEX. In parallel to his clinical work in internal medicine, he studied physics at the University of Munich where he received a German master's degree (Dipl.-Phys. Univ.) in theoretical physics from the University of Munich in 1996 for a scientific thesis on pattern recognition. Since 1997, he has been working as a fellow of radiology at the Department of Clinical Radiology University of Munich where he founded the Digital Image Processing Group. In 2006, he received a Ph.D. degree (Dr. rer. nat.) from the Department of Electrical and Computer Engineering at the Technical University of Munich for a thesis on a novel pattern recognition algorithm invented by him (Exploratory Observation Machine—XOM) for which he holds an international patent. His main research interest is focussed on innovative strategies for pattern recognition and computational intelligence in biomedicine with a specific emphasis on computer-aided diagnosis in biomedical imaging such as functional MRI for human brain mapping and the diagnosis of breast cancer in MR mammography. Dr. Wismller is the author of more than 100 scientific journal and conference publications related to pattern recognition and holds several international patents. He also holds a New York State medical license a German medical board certification in radiology (Facharzt fr Radiologie) and a state doctorate (Habilitation) at the University of Munich. Since August 2007, he has been with the University of Rochester, New York, USA, as Associate Professor for both Radiology and Biomedical Engineering.